AFRL-RI-RS-TR-2008-140
Final Technical Report
May 2008

RATE-ADAPTIVE VIDEO CODING (RAVC)

FastVDO LLC

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-RI-RS-TR-2008-140 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

DAVID HENCH
Work Unit Manager

/s/

WARREN H. DEBANY, Jr.
Technical Advisor, Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAY 08	Final	Jun 06 – Dec 07

4. TITLE AND SUBTITLE
RATE-ADAPTIVE VIDEO CODING (RAVC)

5a. CONTRACT NUMBER FA8750-06-C-0190

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER N/A

6. AUTHOR(S)
P.T. Topiwala

5d. PROJECT NUMBER UAVT

5e. TASK NUMBER GC

5f. WORK UNIT NUMBER DH

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
FastVDO LLC
5840 Banneker Rd
Columbia MD 21044

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/RIGC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TR-2008-140

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# WPAFB 08-3228

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The SmartCapture, H.264 technology for off-processor video encoding with Rate-Adaptive Video Coding (RAVC) extensions is state-of-the-art H.264 at the optimum time in the H.264 development cycle for DoD applications such as UAS video. A USB stick embodiment of the technology is described. This SmartCapture device incorporates what probably is the best available commercial H.264 codec ASIC, a small processor, and an NTSC digitizer. This device leveraged commercial development funds and therefore the technology has not yet been optimized for a particular UAS solution. RAVC scales data rate: 1) in the spatial domain, 2) in the temporal domain, 3) in the encoder fidelity domain, and 4) in the group of pictures (GOP) domain. Useful video can be produced from 32 kbps to 4Mbps. In the spatial domain, image resolutions of 720x480 pixels, 640x480, 352x288, 320x240 and 160x120 can be produced. In the temporal domain, possible frame rates are 30, 15, 10, and 5. In the fidelity domain, at standard resolution, typical rates vary from 2.5 Mbps with appreciable transform artifacts at lower rates. GOP varies noise immunity.

15. SUBJECT TERMS
H.264, SmartCapture, RAVC, UAS video, advanced video encoding AVC, JSCC

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	18	David Hench / N/A

Introduction

The SmartCapture rate adaptive, H.264 video encoder solution for UAS and other systems is believed to be the world's smallest H.264 solution. In its commercial form, this device has an USB memory-stick package-style and USB communications to with the host computer. The full rate adaptivity of the H.264 standard has been made available to the user for on-the-fly changes. This adaptivity is normally only made available to the developer of the numerous products for which the ASIC chip used was designed.

The SmartCapture device incorporates what probably the best available commercial H.264 codec ASIC, a small processor, and an NTSC digitizer. This device leveraged commercial development funds and therefore was not optimized for the UAS solution. A better solution would use a larger processor, which could perform other UAS functions such as communications, autopilot, and scan of a CCD video chip. A block diagram is shown below in figure 1.

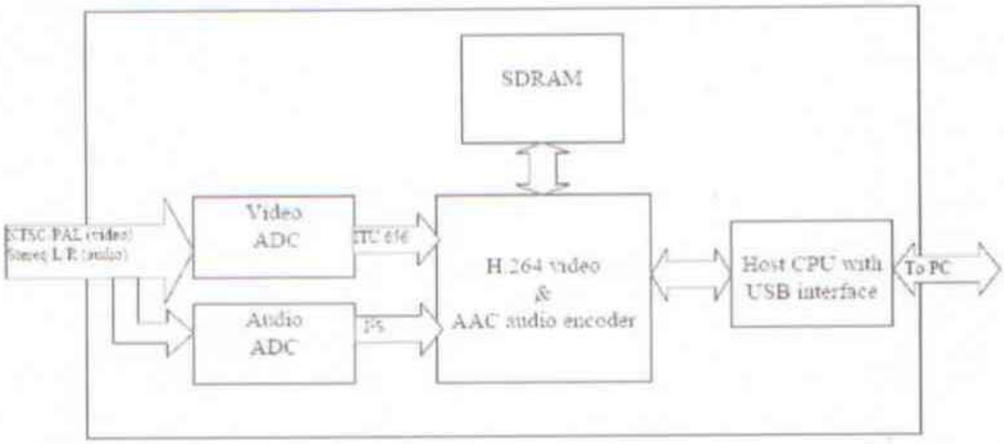


Figure 1. Block diagram of SmartCapture device

This device allows for flexible joint source-channel coding, with data rate adapting to the channel. It also allows a pre-look at what will be possible with the just released Scalable Video Coding (SVC) standard (ITU/H.264 | ISO/IEC MPEG-4 Part 10, Amendment 2). However, a key difference is that SmartCapture currently scales the data rate of the video only at the source whereas, the scalable standard produces a bitstream where the data rate (and quality of video)can be changed at any place in the communications chain by throwing away preselected packets.

Networking software with RTP/RTCP, channel estimation, and adaptive rate control has also been developed and is described here.

Rate adaptivity can be done with NTSC video by scaling: 1) in the spatial domain, 2) in the temporal domain, 3) in the encoder fidelity domain, and 4) in the group of pictures (GOP) domain. Useful video can be produced from 32 kbps to 4 Mbps. In the spatial domain, image resolutions of 720x480 pixels, 640x480, 352x288, 320 X 240 and 160 x 120 can produced. In the temporal domain, frame rates of 30, 15, 10, and 5 can be used. In the fidelity domain at standard resolution typical rates vary from 2.5 Mbps to 500 Mbps with appreciable transform artifacts at lower rates. A typical GOP at 2.5 Mbps may be 30 frames; lower GOP values produce more noise immunity and higher data rates.

Five points of view can be taken on the results of this development

1. It is a commercial product and marketing strategy. From the standpoint of amortizing the research the product is a loss leader for subsequent product development.
2. It is a device for evaluation in battlefield video sensors including UASs.
3. It is a rapid prototyping facility for what is possible with H.264 video.
4. It is a hardware video source solution for Joint Source-Channel Coding.
5. It is a prelook at the emerging Scalable Video Coding (SVC) standard

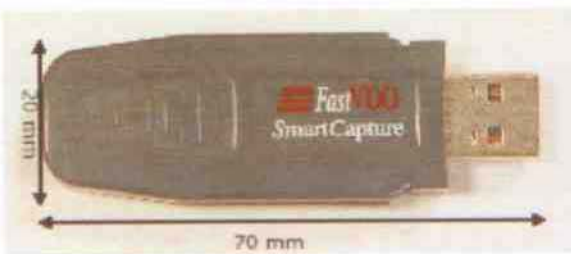
The initial research was performed under an AFRL Dual Use Program on Joint Source-Channel coding. The development of the device itsself was co-funded by Navy Navair for potential use in FireScout UAV, AFRL/RIGC, and commercial funds. SmartCapture is under evaluation by US Army Redstone arsenal, including leadership of Apace Helicopter. SmartCapture is also being tested and/or used by a number of defense companies; including SAIC, Booz Allen, Harris, Echo, as well as the NCIIF (Network Centric Interoperability and Integration Facility) at AFDRL/RIGC. SmartCapture has also been presented to the Motion Imagery Standards Board. During development a presence was maintained both by the contractor and AFRL/RIGC on the JVT (Joint Video Team) between ISO/MPEG and ITU/VCEG as a member of the US National Body of MPEG on the scalable video addition to H.264 video standard. (As a side note, Dr. Topiwala, President of FastVDO, has been the Treasurer of the US National Body since March, 2004, and is currently running for Chairman of the US National Body.)

The remainder of this report shows the technical specs of the SmartCapture device, the manual for the device, the manual for the commercial software and a user's manual for the networking software developed for AFRL. The interested reader is referred to W. Dai; Sachin Patil; Pankaj Topiwala; David Hench, *Rate Adaptive Live Video communications over IEEE 802.11 Wireless Networks, SPIE* Proceedings 6696, Sept 2007.

Appendix A. SmartCapture Specifications

FastVDO SmartCapture – For Video Sensor Data Acquisition

System integrators, military agencies, Satellite and internet bandwidth providers are faced with the challenge of compressing and delivering high quality audio and video streams over constrained bandwidth links. FastVDO's SmartCapture encoder is designed to capture audio /video data with H.264/AAC compression standards to deliver high quality video and audio at low bit rates. While having a portable form-factor and using very low power, SmartCapture allows encoding at various bitrates and interfaces with a broad spectrum of AV sources and has a convenient USB interface to control and transfer compressed data into a PC/ laptop/single board computer for storage and live streaming. SmartCapture design is flexible and customizable to various interfaces and connectors.



• Low latency, real-time capture and encoding
Ideal for situational awareness, surveillance, unmanned systems, reconnaissance and robotics.

• Low power, light weight, small form factor
Smaller and lighter payload to accommodate multiple sensors makes SmartCapture efficient and ideal for power and space limited applications. Weighs < 1/2 ounce, runs on < 1 watt.

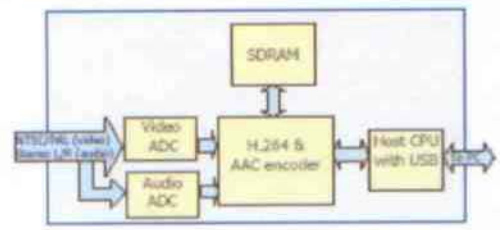
• Powerful onboard tools
Digitizing, resizing, pre-processing and compression is all done on hardware to minimize latency and improve system performance.

• Rugged design and convenient interface
Well engineered design and USB interface makes SmartCapture plug and play on various computing systems and agnostic to OS.

• Fully configurable
SmartCapture encodes video from 32 kbps up to 4 Mbps with tight rate control for varied applications and bandwidths. FastVDO provides APIs to easily configure and control the device. The interface can be customized per application.

• Metadata support and Standards compliant
Compliance to H.264 and AAC for interoperability. Support for CE-608 (Line 21 analog), and SMPTE 336 (KLV metadata).

FastVDO SmartCapture: System Overview



Features	Specifications
Input Video	Analog NTSC or PAL
Input Audio	Analog stereo (L/R)
Output Video	H.264 compliant stream
Output Audio	AAC compliant stream
Bitrate	From 32 kbps up to 4 Mbps
OS supported	Windows, Linux, Mac OS
Frame size	From 160X112 up to 720X576
Power source	5V USB powered
Power	Less than 1 Watt
Weight	0.3 oz. / 9 gm
Dimensions	0.4X0.8X2.8(inch)/ 10X20X70 (mm)

7150 Riverwood Dr. Columbia, MD 21046. Tel: 410-309-6066. Fax: 410-309-6554. Web: www.FastVDO.com E-mail: contact@fastvdo.com

Appendix B. Paper Published in SPIE 2007

1. OBJECT AND SCOPE

Video is increasingly available on the battlefield due to the advent of small wireless cameras. These cameras can be mounted in man-portable UASs or in other battlefield sensors. Battlefield communications are becoming increasingly net-centric and video and raw, unedited and unprocessed video will not fit the net-centric model since the uncompressed bandwidths are too high. Currently, bandwidth constraints prevent this video from being effectively shared with battle commanders or with intelligence operatives. In the battle field scenario, the viewer is trained and he could help in the trading off the video quality he required in order to select and view more video. These factors make the rate adaptive and scalable video essential [1].

The scope of this effort is to develop real-time rate adaptation in video coding with link-quality analysis. Include the ability to scale down in rate, resolution, and frame rate.

2. INTRODUCTION

This report presents an example rate adaptive, live, video communication system over IEEE 802.11 wireless networks, which integrates channel estimation, rate adaptive video encoding, wireless transmission, reception, and playback. This system has the following features: 1) IEEE 802.11 wireless WLAN is used as the communication network; 2) live video capturing and rate control coding via an external USB device; 3) low power consuming electronics; 4) realtime bandwidth estimation; 5) video streaming conforming to RTP and RTCP protocols.

In most video streaming applications, the video sources are pre-existing (e.g., film content). In such cases, a streaming application uses the bandwidth estimation to adapt the sending rate only. In our application, the bandwidth estimation changes the video coding rate as well and hence the video quality adapts to available bandwidth. A rate adaptation mechanism will be used between bandwidth estimation and rate control to track the dynamic bandwidth changes and guarantee smooth and stable video encoding.

Rate adaptivity is an important concept in data intensive applications such as video communications. The available bandwidth to a live video application will directly impact the application's performance, i.e. resolution, quality, frame rate, bitrate and delay. Wireless networks impose extra challenges to bandwidth estimation because they are very sensitive to environmental conditions including low reception signal strength, path loss, fading, interference or connectivity itself. Those effects become more pronounced when the platform itself is in motion. Live video communication over quick transition wireless channel needs to update the available bandwidth every few seconds to avoid client-side buffer underflows and to limit user wait periods to use the application. This implies a fast convergence time requirement. The bandwidth estimation is typically taken within a single application stream. This adds one more requirement that a bandwidth estimation tool must be minimally intrusive so as to not adversely impact the application's performance during measurements.

Although bandwidth estimation techniques have been widely studied recently [2,3,4,5,6 and reference therein], many bandwidth estimation techniques are developed for wired networks and most bandwidth estimation tools are aimed to network management and monitoring instead of real time video applications. Accuracy is usually the main concern when comparing different techniques or tools. Live, wireless video communication poses different requirements to the bandwidth estimation. Realtime update becomes the key challenge. Accuracy therefore, although desirable, is no longer the primary concern. In this report, we introduce a method, which addresses wireless network properties yet remains practical for live video streaming.

The remainder of this report is organized as follows. Section 2 summarizes measurement metrics and existing channel estimation techniques. Section 3 describes the system infrastructure of the rate adaptive, live, video communications system over IEEE 802.11 wireless network and the functions of each module. Section 4 describes the rate adaptive live video communication algorithm. Experimental results are presented in section 5. Section 6 gives conclusions and suggestions for future work.

3. REVIEW OF BANDWIDTH ESTIMATION TECHNIQUES

This section reviews existing measurement techniques for capacity and available bandwidth.

The capacity C and available bandwidth A are two commonly used metrics, as they relate to the amount of data that a link or network path can deliver per unit of time. The capacity is the maximum IP-layer throughput that the path can provide to a flow, when there is no competing traffic load (cross traffic). The available bandwidth, on the other hand, is the maximum IP-layer throughput that the path can provide to a flow, given the path's current cross traffic load to digital communications [4].

Let H be the number of hops in a path, C_i be the transmission rate or capacity of link i, and C_0 be the transmission rate of the source, then the end to end capacity C is determined by the link with minimum transmission rate [4]:

$$C = \min_{i=0,\ldots,H} C_i \ , \tag{1}$$

If u_i is the utilization of link i (with $0 \leq u_i \leq 1$ and $u_0 = 0$), the available bandwidth A of the path is determined by the link with the minimum unused capacity [4]:

$$A=\min_{i=0,\ldots,H} [C_i(1-u_i)], \tag{2}$$

Current active bandwidth estimation techniques can be divided into 4 categories: 1) single packet probing, like variable packet size; 2) packet pair/train dispersion probing; 3) self-loading periodic streams; 4) probe gap model techniques.

Variable packet size measures the round trip time RTTs to all hops on the path. Jacobson's pathchar [9], Downey's clink [10], Mah's pchar [11] use this technology to measure the capacity for every link in a path [4]. These tools usually require ICMP replies from the router. The measurement is often quite inaccurate as different routers may have different ICMP response times. Certain network elements may not even have an ICMP response.

Self-loading periodic streams send a number of equal-sized packets to the receiver at certain rate R. Train of Packet Pairs (TOPP) [12], pathload [13] and path Chirp [14] probe the end-to-end network path using increasing probing rate. They provide accurate bandwidth estimation for wired network at the cost of long convergence time and high intrusiveness [6].

Probe Gap Model techniques, such as Initial Gap Increase/Packet Transmission Rate (IGI) [15], measure available bandwidth by estimating the crossing traffic at the tight link and by monitoring the gap changes after the packets pass through the tight link router. However, IGI assumes a known constant capacity, which is not valid for wireless networks.

Packet dispersion techniques [4 and references therein], including packet pair and packet trains, measure the end-to-end capacity of a network path. Packet pair dispersion sends two equal-sized packets back-to-back into the network. After traversing the narrow link, the time dispersion between the two packets is linearly related to the link with the least capacity. Packet train dispersion extends packet pair dispersion by using multiple back-to-back probing packets. However, the concepts for packet train are similar to that of a single packet pair.

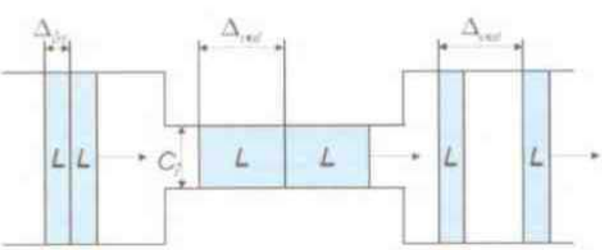


Fig. 1. Packet Dispersion. The width of each link corresponds to the capacity [4]

Among these methods, packet dispersion is one of the most simple and mature bandwidth estimation techniques for wireless networks, thanks to its fast measurement time and modest network load.

Fig. 1 illustrates the packet dispersion concept. When packets of size L with initial dispersion Δ_{in} go through a link of capacity C_i, the dispersion after the link Δout becomes [4]:

$$\Delta_{out} = \max(\Delta_{in}, \frac{L}{C_i}) \tag{3}$$

After packets traverse each link on an H hop end-to-end path, the final dispersion at the receiver is [4]:

$$\Delta_R = \max_{i=0,\cdots,H}(\frac{L}{C_i}) = \frac{L}{\min_{i=0,\cdots,H} C_i} = \frac{L}{C} \tag{4}$$

Where, C is the end-to-end capacity. Therefore, the end-to-end path capacity can be estimated by $C=L/\Delta_R$.

In wired network, the capacity of link C_i is assumed to be a fixed value. In wireless network, special features, such as dynamic rate adaptation, random delay between successive packets, MAC layer connection backoff, MAC layer ARQ, basic two-way handshake or four-way handshake influence the capacity instantly. M Li, M Claypool and R. Kinichi proposed two packet dispersion measurements: effective capacity C_e and achievable throughput A_t to emphasis these significant differences between the wired and wireless network. Since transmitting rate in wireless WLAN is time varying, effective capacity C_e is defined as a function of the packet size and time used to indicate the effective transmit rate of the wireless network to deliver network layer traffic during a given time period [6]. Replacing Δ_R by a continuous variable T(t) and take average over the given time period $[t_0, t_1]$, $C=L/\Delta_R$ can be extended to wireless situation:

$$C_e = \frac{\int_{t_0}^{t_1} \frac{L}{T(t)} dt}{t_1 - t_0} \tag{5}$$

Where, T(t) is the packet pair dispersion at time t.

Given discrete packet pair samples T(i) (Δ_R at time index i), the average effective capacity C_e is [6]:

$$C_e = \frac{\sum_{i=1}^{n} \frac{L}{T(i)}}{n} \tag{6}$$

Achievable throughput A_t is the maximum throughput that a node can achieve in contending with other existing traffic in a wireless network. This is approximated as the length of the packet divided by the average dispersion [6]:

$$A_t = \frac{nL}{\sum_{i=1}^{n} T(i)} \tag{7}$$

Where n is the number of samples from packet pair measurements and T(i) is the dispersion of the nth packet pair.

In this report, we will use capacity or available bandwidth if not specified otherwise. But the user should keep the difference mentioned above in mind.

4. SYSTEM INFRASTRUCTURE

The rate adaptive video communications system contains two parts: the server and the client receiver. The system flow graphs of both parts are given in Fig. 2. The system setup is illustrated in Fig. 3.

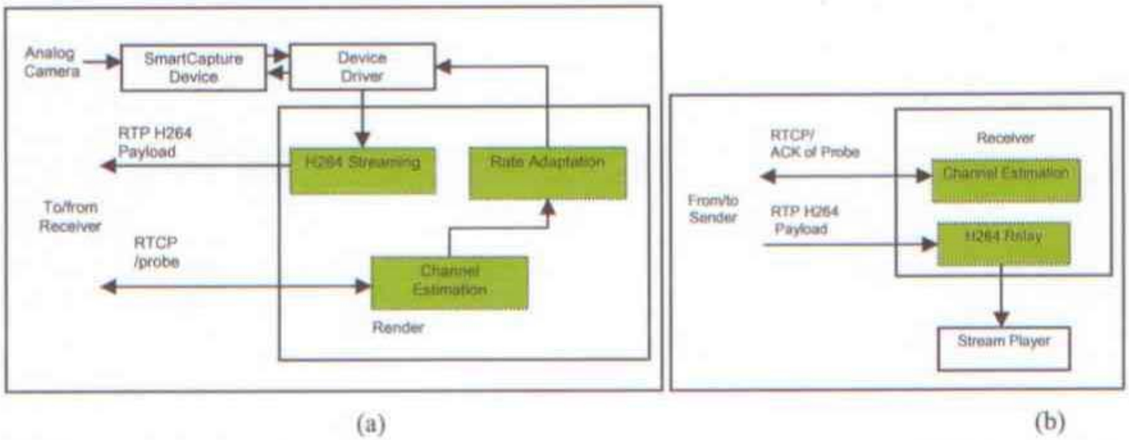


Fig. 2. System flow diagrams: (a) rate adaptive video encoding streamer; (b) rate adaptive video receiver

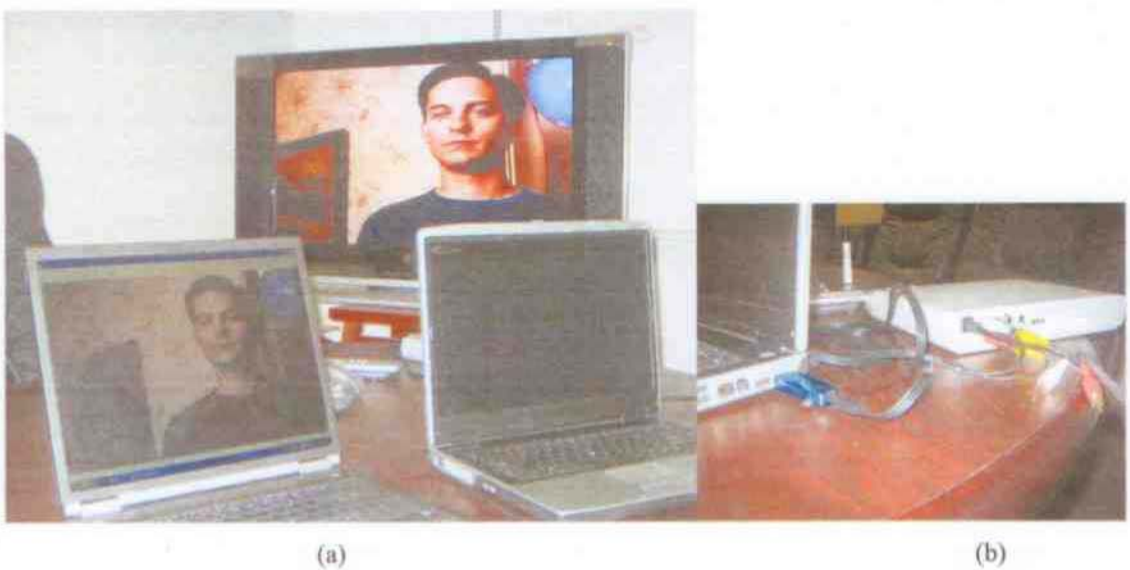


Fig. 3. (a) An example system setup of the rate adaptive video communication system; (b) A close-up look of the SmartCapture device hooked to the USB port of the transmitting system

In Fig. 3, the laptop on the left is the receiver. The laptop on the right is the transmitter. SmartCapture device is hooked to the USB port of the transmitter laptop. The output of a DVD player is split to SmartCapture as well as a large screen. Video is transmitted via a wireless network. QuickTime is used to play the received video stream. The video also display to the LCD simultaneously. Note the near simultaneity of the live playback on the big screen, and the captured, encoded, transmitted, received and decoded digital signals.

4.1 Video Encoding

Video encoding is done by the SmartCapture device shown in Fig. 4. The SmartCapture device is an external USB device which converts the analog video and audio capture from VCR, camcorder or video camera to H.264/AAC/MP4 stream and connects the stream to the PC or Laptop. It is today the world's smallest H.264 SD encoder board. Other similar products are available (e.g., Hauppage USB-Live, http://www.hauppauge.com/pages/products/data_usblive.html) for MPEG-2 and MPEG-4, but no other product has been announced for H.264/AVC.

Fig. 4. FastVDO SmartCapture device

On-board chips digitize and down-sample (if needed) the analog video signal, then compress the video into the H.264 format. H.264 baseline profile is used for both video encode and decode. The device can be optimized for either CBR or VBR rate control. The frame rates currently support are 30/15/10/5 frames per second. Commands can be send from the PC to the device via the USB port to change bitrate on the fly. Despite the bitrate, other video coding parameters such as resolution (width and height), frame rate and GOP (group of picture) can also be changed if necessary. For more information about the SmartCapture device, please visit http://www.fastvdo.com.

4.2 Bandwidth Estimation

The channel estimation module is a crucial part of the whole system. If the available bandwidth is overestimated, more data will be sent to the network than can be handled, and cause congestion. If the available bandwidth is underestimated, the video quality suffers from being compressed at a lower bitrate (or framerate) than necessary. Of course, overestimation is much more harmful than underestimation. The packet loss rate increases dramatically when congestion occurs, which puts more stringent requirements for error concealment and error robustness at the decoder side.

Small disruption of the streaming video is another requirement for the bandwidth estimation algorithm. Probes (UDP packets) are used to sniff the network traffic. Generally speaking, the more probes are sent, the more accurate the estimated bandwidth. However too many probes can possibly delay or even block the video streams. Ideally, the probes should take very little bandwidth or cause no intrusion, meaning there is no interference to the video stream.

Speed, which is mainly determined by the algorithm convergence time, is also an important consideration. Environmental conditions generally cause wireless capacity variability to occur randomly over short times. Bandwidth estimation should keep up with the change. Wireless communications between UAVs and ground station, for example, is more susceptible to the outdoor environment and rapidly changing distances. Accurately algorithm with long converge time is not acceptable here since it does not satisfy real time available bandwidth update. A video friendly available bandwidth estimation algorithm will be described in section 5.

Accuracy therefore, although desirable, is no longer the primary concern. The streaming media applications change the sending rate in quantum steps instead of doing so smoothly. For example, in **Table 1**, any bandwidth estimated between 128Mbps to 256Mbps will trigger the device to code at 128Mbps. While finer rate adaptation can be achieved in a slowly changing environment, it may not be useful in a rapidly changing environment.

4.3 Rate Adaptivity

The estimated available bandwidth is calculated from the statistics at the current instance or for the past duration. It forecasts but doesn't guarantee that the real available bandwidth during the following few seconds is exactly the same. It can drop quickly and dramatically, for example upon observation of massive cross traffic or because of uncertain channel fading. For these reasons, the estimated available bandwidth cannot be used directly to change the video coding bit rate at the newly estimated rate. Instead, we increase the video bit rate step by step slowly in case the estimated available bandwidth increases. On the other hand, if the estimated available bandwidth decreases, the video bit rate should be dropped immediately to reflect the change. Table 1 gives an example how to change video bitrate R observing the available bandwidth A. **Table 2** shows current default video coding parameters given the video bitrate R.

4.4 Video streaming

The standard Real-time Transport Protocol (RTP) builds on UDP/IP, and provides timing recovery and loss detection, to enable the development of robust systems. There are two parts of RTP: the data transfer protocol and an associated control protocol (RTCP), which is used for periodic reporting of reception quality, source description information, and the information needed to synchronize media streams [7,8]. We have used RTCP packets at 1 packet per second to communicate from the streamer to the receiver, sender report (SR), and back, receiver report (RR) to communicate packet loss and throughput.

Table 1. Video bitrate (R) adjusted according to previous video bitrate (B) and available bandwidth (A)

B Kbps	0<A<64	64<A<128	128<=A< 256	256<=A< 512	512<=A< 1024	1024<=A< 1536	A>=1536
64	64	64	128	128	128	128	128
128	64	64	128	256	256	256	256
256	64	64	128	256	512	512	512
512	64	64	128	256	512	1024	1024
1024	64	64	128	256	512	1024	1536
1536	64	64	128	256	512	1024	1536

Table 2. Video coding parameters changed with bitrate

B	64Kbps <=B< 200Kbps	200Kbps<=B< 400Kbps	400Kbps<=B< 1.5Mbps	B>1.5Mbps
Resolution	160x120	320x240	320x240	640x480
Frame rate	15	15	30	30
GOP	15	15	30	30

5 LIVE RATE ADAPTION ALGORITHM

Since packet dispersion is the most simple and efficient method to give quick bandwidth estimation, we will use this technique in our application. A variable length packet train of UDP packets without RTP encapsulation will be sent to the receiver to estimate the available bandwidth. The size of each probe packet is 1500 bytes, or 12000 bits. The length of the packet train, however, is determined by the previous estimated available bandwidth such that 5% of the available bandwidth is used for probing. When the available bandwidth is extremely low, no probe will be sent until the application decides it is time to increase the video coding and sending rate, which for example, can be triggered by no video packet loss for several seconds. Let subscript i indicate the time index with a one second interval, given the available bandwidth at time i, A_i, the length of packets N at time i is:

$$N_i = \lceil 0.05 * A_{i-1} / (1500 \times 8) \rceil = \lceil A_{i-1} / 240000 \rceil, \text{ for } i = 1,2,3\ldots \quad (8)$$

The achievable throughput is calculated by the receiver as $A_i = n*L/(\Sigma_{j=1,\ldots,n} T_j) = 12000n/(\Sigma_{j=1,\ldots,n} T_j)$, where L is the probe packet size in bit and $n <= N_i$. When part or all packets are lost, n is less than N_i. The T_i is the time dispersion between packet i-1 and i, for i=1,2,...,n, which can be measured by recording the arriving time of packet i-1 and packet i.

Although fast and minimal intrusion available bandwidth can be estimated by packet dispersion method used in our application, it can over estimate the available bandwidth. For example, when the length of packet train N is small, it cannot always capture the cross traffic in the network thus the estimation is higher than the actual value. This overshooting can be avoided by take the receiving video data rate into consideration. If the receiving video data rate is less than the estimated bandwidth, the smaller value will be used. The receiving video data rate can be calculated from the RTCP sender report (SR) and receiver report (RR). Let's define $R_{snd,i}$ to be the video sending bit rate between two consecutive RTCP SR at time i and i-1, $R_{rcv,i}$ to be the video receiving bit rate received by the receiver, and L_i to be the loss fraction at time i, which will be available from RTCP RR at time i. Then we have $R_{rcv,i} = R_{snd,i} * L_i$. R_{snd} can be easily calculated by the sender.

Other terms used in this session are defined here. B_i is the device coding bitrate at time i. The live rate adaptive algorithm is summarized as follows:

Step 1: <u>Initialization:</u>

1.1 Set $A_0 = 1$ Mbps, set $N_0 = \lceil A_0 / 240000 \rceil = 5$, set $R_{rcv,i} = R_{snd,i} = L_i = 0$, $B_0 = 512$Kbps

Step 2: <u>Packet loss detection:</u>

2.1 Set $L_i = 0$

2.2 Sender sends RTCP SR_i and updates $R_{snd,i}$

2.3 Receiver receives SR_i and responses with RR_i

2.4 If sender gets RTCP RR_i,

2.4.1 Update L_i from RR_i

2.4.2 Calculate $R_{rcv,i} = R_{snd,i} * L_i$

2.4.3 Set SendProbePackets = true

2.5 Else if sender cannot get RTCP RR_i

2.5.1 Set $R_{rcv,i} = 0$

2.5.2 Set $A_i = A_{i-1} / 2$

2.5.3 Set SendProbePackets = false

Step 3: <u>Achievable throughput estimation:</u>

3.1 If SendProbePackets

3.1.1 Calculate $N_i = \lceil A_{i-1} / 240000 \rceil$, set $A_i = 0$

3.1.2 Send N_i probes to receiver

3.1.3 Calculate A_i using equation (7)

3.1.4 Receiver sends back A_i to sender

3.1.5 If sender gets feedback, update A_i

Step 4: <u>A_i validation:</u>

4.1 If $L_i > 0$, $A_i = \min(A_i, R_{rcv,i})$

Step 5: <u>Video rate adjustment:</u>

5.1 Update B_i according to Table 1

5.2 Send change parameter command to SmartCapture device according to **Table 2**

6 EXPERIMENTAL RESULTS

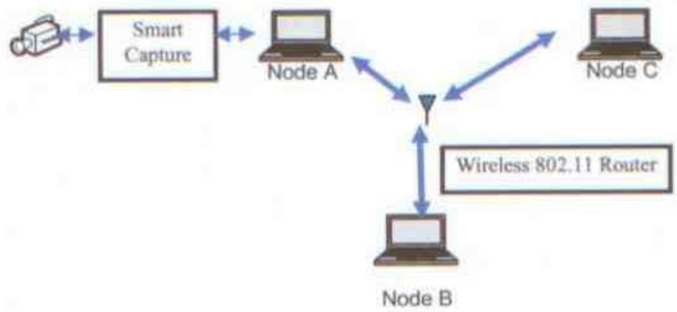


Fig. 5. Lab setup

Three computers, labeled as A, B and C shown in Fig. 5 are wirelessly connected via an 802.11g network. Node A serves as the video streamer. Node C is the video receiver. Node B is used to add load to the network. The operation system Node A uses is open SUSE 10.2 with 2.6 Linux kernel. The wireless network card used in node A is Intel PRO/wireless 2200BG. The operation system on Node B and Node C is Windows XP, the wireless adaptors used in node B and C is D-Link DWL-G122. The router we use is Dlink DI-524, which supports both 802.11 b and g. We use iperf [17] to add load and as an external, third party bandwidth estimation tools. The network traffic is categorized as: video stream (from A to C), RTCP and probe (two-way between A and C) and cross traffic (sending by iperf from B and C).

6.1 Estimate the available bandwidth with no load

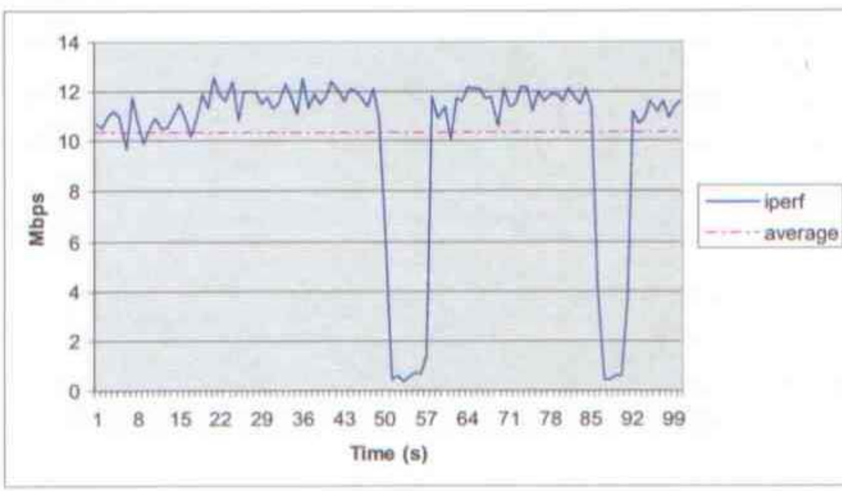


Fig. 6. Available Bandwidth as estimated by a popular tool called iperf. Test duration is 100 seconds

The popular tool iperf [16] is used to estimate the bandwidth and add load to the network. Iperf uses bulk data transfer method to saturate the channel. In Fig. 6, ipef is used only to measure the effective capacity. The capacity is not a constant in this test because it is very sensitive to environmental conditions including low reception signal strength, path loss, fading, interference of other wireless devices or connection itself. Further more, those conditions could possibly trigger the rate adapt algorithm embedded in the router and make it switch to a lower transmitting rate. That can explain the sudden drops on the estimated available bandwidth. Sometimes, it drops dramatically. The test lasts for 100 seconds. The average capacity over the 100 seconds is 10.3 Mbps.

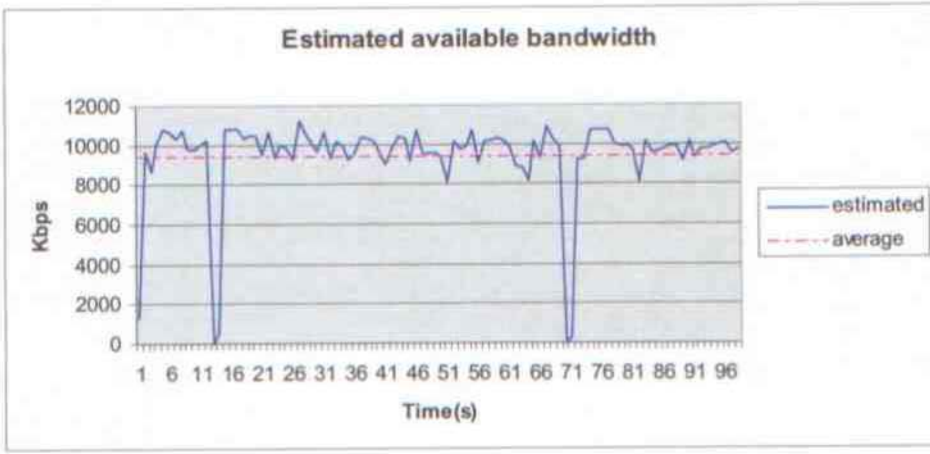


Fig. 7. Available bandwidth as estimated by FastVDO. Test duration is 100 seconds

Fig. 7 shows the result estimated by our application. No other traffic present except the probes. The test lasts for 100 seconds with an estimated capacity around 9.4Mbps. This is about 0.9 Mbps less than as estimated by iperf. Note that unlike iperf, which uses full bandwidth in its bulk data transfer, FastVDO's method uses only about 5% of the available bandwidth. This approach is much more traffic friendly. A running streaming application cannot afford to run iperf, or indeed any intrusive bandwidth estimation approach which uses a significant fraction of the bandwidth for measurement. Also, FastVDO's approach is real-time, in which estimation time is less than 30 ms.

6.2 Estimate the available bandwidth with and without load

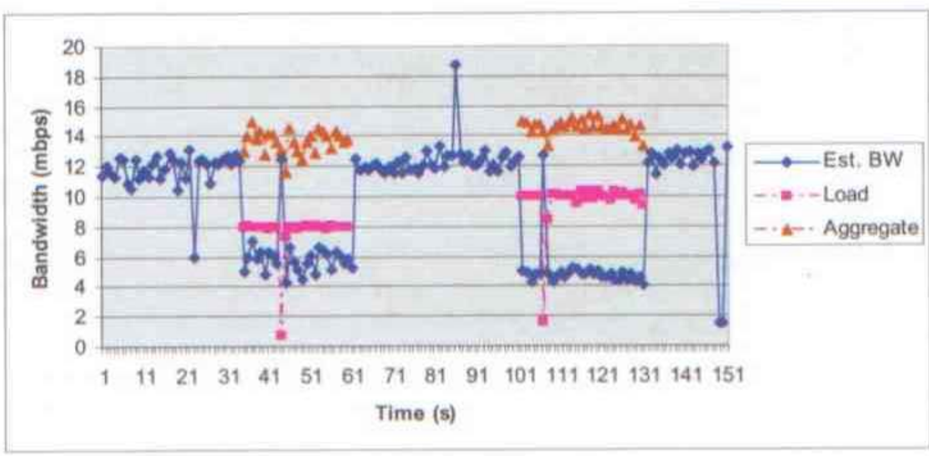


Fig. 8. Aggregate bandwidth with injected load and estimated available throughput

In Fig. 8 the test duration is 150 seconds. The square purple line is the load added to the network by iperf. During 31-60 seconds, the load is 8 Mbps. During 101-130 seconds, the load is 10 Mbps. The diamond blue line indicates the estimated achievable throughput by FastVDO. The triangle yellow line is the aggregated bandwidth.

6.3 Rate adaptive video streaming with and without load

In this section, load will be added to the network to investigate the adaptivity of our application. Available bandwidth is updated every one second, which is believed to be adequate to the system real time update requirement.

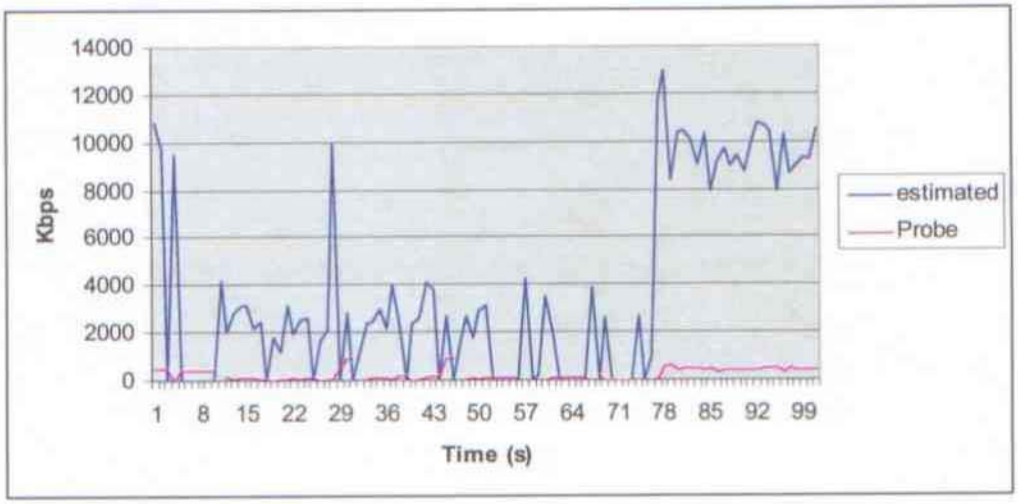


Fig. 9. Estimated available bandwidth and the bandwidth used by probing

Fig. 9 illustrates that our method is nonintrusive. In Fig. 9, cross traffic is added to make the available bandwidth have a large dynamic rage. This figure shows the bandwidth used by probing is changing with the previously estimated available bandwidth. It is about 5% of the available bandwidth. When the bandwidth is extremely low and packet loss is detected, no probes will be sent. All the available bandwidth will be used by the video data. The "Probe" curve shows the actual probe data rate.

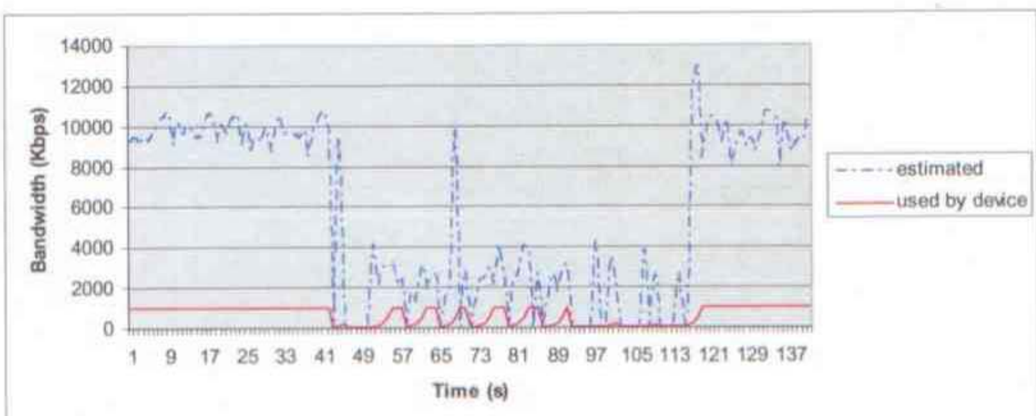


Fig. 10. Estimated available bandwidth and SmartCapture device bitrate settings

In Fig. 10, various cross traffic are added to the network from 41 second to 120 second to test the behavior of our rate adoption algorithm. The dotted line is the estimated available bandwidth. The solid line is the device settings. Whenever the bandwidth is drop, the device coding rate corresponds immediately by jumping down to a smaller value. The device rate recovers one step up each time. For example, from t= 96 to 114 second, the bandwidth oscillates frequently. Ambitious rate increasing will make the network congestion even worse. And aggressive rate change will cause the rate control algorithm embed in the device work inefficiently. Our application uses smooth rate increase so that the device has a smooth transition. Sharp rate deduction is used so that the data sending rate is always below the available bandwidth.

Fig. 11 shows the visual quality at different bitrate, framerate and group of picture (GOP). The input is the NTSC analog signal. The video is then down sampled to 320x240. The first frame is coded as I frame. Although we can tell difference for the grass, the visual quality of the main objects, such as vehicles, building and trees in each picture are reasonably consistent and acceptable. That is the advantage of reduce framerate at low bitrate. Considering that UDP is a unreliable transfer protocol, the GOP value changes with framerate so that we can always refresh the I frame every one second to reduce error propagation. The resolution can also be changed if necessary. However, frequent change of resolution is not suggested. If the sequence parameter set (sps) and picture parameter set (pps) packets are lost, the decoder could not be informed of the resolution change, which will possibly make the decoder crash. Possible solution is send sps and pps over a reliable connection.

7 CONCLUSION

In this report, effective capacity and achievable throughput are used as metrics to measure the available bandwidth of the wireless network. We use a packet-dispersion-based algorithm to estimate the available bandwidth. The dispersion rate may be larger than the available bandwidth if the number of packets is not big enough to capture the cross traffic. To overcome the overestimation, this algorithm uses the information such as loss fraction and receiving data rate from RTCP packet to validate the estimation. This realtime algorithm is no intrusion, thus video friendly. Video encoding is done by the external USB SmartCapture device, thus the system requirement for the streamer is low. For example, the CPU usage is around 1% in the system with Intel Pentium M 2.0G CPU. By design the whole system is easy to migrate to mobile devices.

Fig. 11. The first frames for different video coding settings. From top to bottom, from left to right: 512Kbps @ 30fps, GOP = 30, 256Kbps @ 15fps, GOP = 15, 128Kbps @ 10fps, GOP = 10, 64Kbps @ 5fps, GOP = 5